DATA MINING PROJECT REPORT

Yufan Cai
516030910496
Shanghai Jiao Tong University
gluttony@sjtu.edu.cn

Xun Xu
516030910516
Shanghai Jiao Tong University
459657718@qq.com

Yingzhe Luo
516030910507
Shanghai Jiao Tong University
949874702@qq.com

June 30, 2019

1 Introduction

Author: Xun Xu
This project is meant to practice our ability of handling actual business problems about financial transaction.
We divide the project into three parts:Regression based stock price prediction,Feature Generation and Trading strategy based on reinforcement learning

2 Environment Configuration

Author: Yingzhe Luo
a) Python3.5.2(Ananconda 4.2.0)
b) Python Packages: Sklearn, xgboost, featuretools, torch(1.0.1)
c) Dataset: The dataset includes the raw data and the provided features. The raw data is the public data pushed by the market, and each tick pushes a group

3 Task Allocation

Yufan Cai: Task1(Regression used machine learning methods such as RF, GBDT, Adaboost), Task2(Wavelet Analysis and Stacked Auto Encoder)

Xun Xu:Task2(simple methods and Deep Feature Synthesis)

Yingzhe Luo:Task3(Recurrent neural network using long-short term memory)

4 Task1: Regression based stock price prediction

Author: Yufan Cai

4.1 Random Forest

Random forest is a Bagging algorithm based on a decision tree-based learner, but the difference is that the random decision attribute (sub-sampling on the feature) is added to the training process of the RF decision tree. The traditional decision tree chooses the optimal attribute when selecting the attribute of the division. First, from the attributes of the node, K attributes are selected to form a random subset (the class is the Random Subspaces in Bagging, usually K=log2(n)). Then select a rightmost subset from this subset to divide.

Because there is sub-sampling of features in RF compared to the general decision tree, the randomness of the model is enhanced. Although this increases the bias, it is because of the integration effect, the variance is reduced, so this usually takes the whole Get a better model In addition to the normal version of the random forest, we can also construct the limit random forest by using the limit random tree. The difference between the limit random tree and the random tree of the ordinary random forest is that the former does not select the optimal attribute when dividing the attribute, but Random selection (implementation in sklearn is to generate a random threshold for each attribute and then select the optimal threshold in the random threshold) 3. Generation of final prediction results: In the original RF paper, the final prediction result is a simple vote for all prediction results, but in our commonly used machine learning library sklearn, the average of the prediction probabilities of each classifier is taken.

4.2 Gradient Boosting Decision Tree

The tree model can solve the problem of nonlinear features. The tree model does not require feature normalization and unified quantization (that is, both numerical and category features can be directly used in the construction and prediction process of the tree model), and the tree model can be intuitively output. The decision process makes the predictions interpretable. To prevent over-fitting when using the tree model, the sensitivity to data noise is high (predictive stability is poor), and it is difficult to construct the best tree model with training data, so the greedy algorithm used in actual operation is similar. The solution can only find some suboptimal solutions.

The data types of the regression leaf nodes are continuous, while the data types of the classified leaf nodes are discrete. The regression leaf nodes are specific values, and the classification leaf nodes are prediction categories determined according to the training sample category. The leaf nodes of the regression tree return the mean of the "one group" of training data, rather than specific, continuous predictions.

4.3 Adaboost and Xgboost

The working mechanism of the Boosting algorithm is to first train a weak learner 1 from the training set with the initial weight, and update the weight of the training sample according to the learning error rate performance of the weak learning, so that the weak learner 1 learns the training sample point with high error rate. The weight of the object becomes higher, so that these points with high error rates are more valued in the latter weak learner 2. Then, the weak learner is trained based on the training set after adjusting the weights, and the repetition is performed until the number of weak learners reaches the predetermined number T, and finally the T weak learners are integrated through the set strategy to obtain the final strong Learner.

The main advantages of Adaboost are: 1) When Adaboost is used as a classifier, the classification accuracy is very high. 2) Under the framework of Adaboost, various regression classification models can be used to construct weak learners, which is very flexible. 3) As a simple binary classifier, the structure is simple and the results are understandable. 4) It is not easy to have a fit The main disadvantages of Adaboost are: Sensitive to the anomaly sample, the anomaly sample may obtain higher weight in the iteration, which affects the prediction accuracy of the final strong learner.

4.4 Result

In this task, we implemented three algorithm to do this regression work. We predict the price after ten time units, which means the label equals (the future n-th tick's AskPrice1 + the future n-th tick's BidPrice1 - the current tick's AskPrice1 - the current tick's BidPrice1) / 2, the value of n is 10. We first use the origin data and then use the standard data which is standardized.

We used four evaluation criteria:

Explanation variance score: explain the variance score of the regression model. The value range is [0,1]. The more the independent variable can explain the variance of the dependent variable. The smaller the value, the worse the effect.

Mean absolute error: mean absolute error (MAE), used to assess the degree of proximity of the prediction result to the real data set. The smaller the value, the better the fitting effect.

Mean squared error: mean squared error (MSE), which calculates the mean of the sum of the squares of the error between the fitted data and the sample point of the original data. The smaller the value, the better the fitting effect.

R2 score: The judgment coefficient is also the variance score of the regression model. The value range is [0,1]. The closer to 1 is, the more the independent variable can explain the variance variation of the dependent variable. The smaller the value, the worse the effect.

The result is listed in the following table.

Method	Explained variance score	Mean absolute error	Mean squared error	R2 score
Decision Tree	0.099	0.471	1.429	0.099
Adaboost	0.529	1.178	2.749	0.562
Random Forest	0.281	0.434	0.699	0.281

Table 1: Results of origin data

Method	Explained variance score	Mean absolute error	Mean squared error	R2 score
Decision Tree	0.096	0.469	1.393	0.095
Adaboost	0.641	1.407	3.701	0.634
Random Forest	0.391	0.437	0.773	0.392

Table 2: Results of standard data

5 Task2: Feature Generation

5.1 Simple methods

Author:Xun Xu
Importing sklearn python package, we use PolynomialFeatures to process the raw data. We set the degree of the polynomial features as 2. Then we select several dimensions of the feature based on MSE to gain the generated features.

5.2 Deep Feature Synthesis

Author: Xun Xu
Featuretools is a framework to perform automated feature engineering. It excels at transforming temporal and relational datasets into feature matrices for machine learning.
An EntitySet is a collection of entities and the relationships between them. They are useful for preparing raw, structured datasets for feature engineering. While many functions in Featuretools take entities and relationships as separate arguments, it is recommended to create an EntitySet.
First, we initialize an EntitySet.To get started, we load the transactions dataframe as an entity.We want to relate these two entities by the columns called "id" in each entity. Each product has multiple transactions associated with it, so it is called it the parent entity, while the transactions entity is known as the child entity. When specifying relationships we list the variable in the parent entity first.Finally, we are ready to use this EntitySet with any functionality within Featuretools.

5.3 Wavelet analysis and Stacked Auto Encoder

Author: Yufan Cai

5.3.1 Wavelet analysis

Wavelet analysis is a relatively new area of signal processing. A wavelet is a mathematical function that decomposes data into different frequency components, and then each component is studied at a resolution that matches its proportion, where the scale represents the time range. Wavelet filtering is closely related to the variability and time-varying characteristics of real-world time series, and is not limited by the assumption of stationarity. Wavelet transform

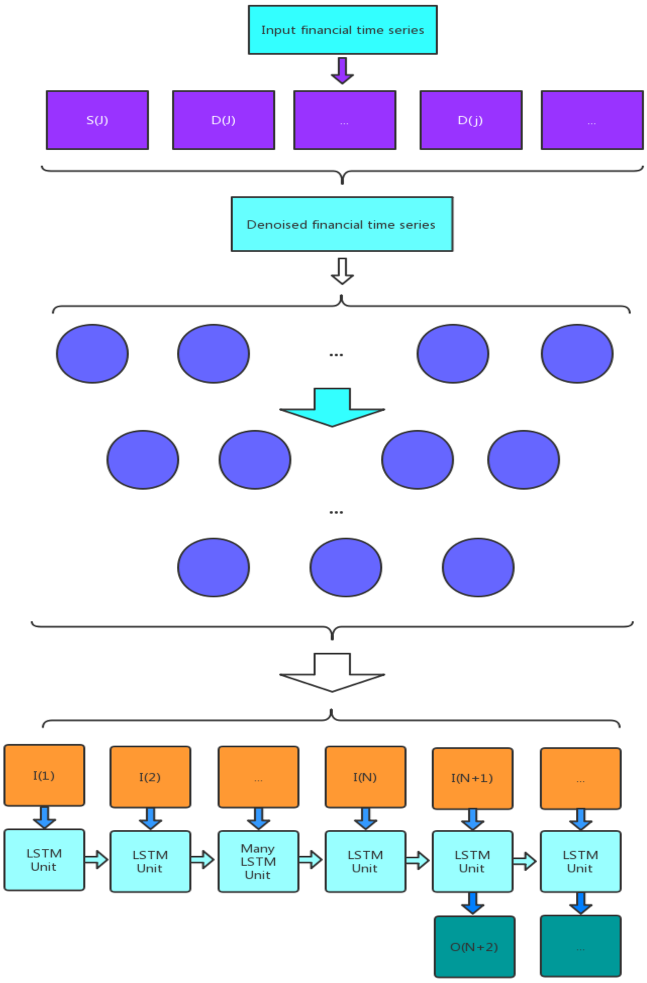

Figure 1: Model framework

decomposes a process into different scales, which helps to distinguish seasonality, reveal structural fracture and volatility clustering, and identify the local and global dynamics of the process on a specific time scale. Wavelet analysis has proven to be particularly useful in analyzing, modeling, and predicting the behavior of financial instruments, such as stocks and exchange rates. In this study, Haar wavelet decomposition wavelet transform is used to decompose time series.

5.3.2 Stacked autoencoders

Stacked autoencoders are built by stacking a series of single-layer autoencoders layer by layer. The single layer autoencoder maps the input daily variables to the first hidden vector. After training the first single layer autoencoder, the reconstructed layer of the first single layer autoencoder is removed and the hidden layer is retained as the input layer of the second single layer autoencoder. In general, the input layer of the subsequent AE is the hidden layer of the previous AE. It is worth noting that the weights and deviations of the reconstructed layers after the completion of training each single layer AE are discarded. In this work, the number of input daily variables for each data set ranges from 64 to 32; then, the size of the hidden layer is set to 10 by trial and error. Depth plays an important role in SAE because it determines the invariance and abstraction of the extracted features. In this work, the depth of the SAE is set from 3 to 5.The model framework can be seen in Figure. 1.

5.3.3 Result

In this task, we implemented four algorithms to do this regression work. We predict the price after ten time units, which means the label equals (the future n-th tick's AskPrice1 + the future n-th tick's BidPrice1 - the current tick's AskPrice1 - the current tick's BidPrice1) / 2, the value of n is 10. We first use the origin data and then use the standard data which is standardized.

We used four evaluation criteria:

Explanation variance score: explain the variance score of the regression model. The value range is [0,1]. The more the independent variable can explain the variance of the dependent variable. The smaller the value, the worse the effect.

Mean absolute error: mean absolute error (MAE), used to assess the degree of proximity of the prediction result to the real data set. The smaller the value, the better the fitting effect.

Mean squared error: mean squared error (MSE), which calculates the mean of the sum of the squares of the error between the fitted data and the sample point of the original data. The smaller the value, the better the fitting effect.

R2 score: The judgment coefficient is also the variance score of the regression model. The value range is [0,1]. The closer to 1 is, the more the independent variable can explain the variance variation of the dependent variable. The smaller the value, the worse the effect.

The result is listed in the following table.

Method	Explained variance score	Mean absolute error	Mean squared error	R2 score
Decision Tree	0.558	0.0.345	0.597	0.558
Adaboost	0.207	0.895	1.657	0.227
Random Forest	0.663	0.337	0.455	0.663
Xgboost	0.105	0.565	1.209	0.104

Table 3: Results after Wavelet analysis

Method	Explained variance score	Mean absolute error	Mean squared error	R2 score
Decision Tree	0.5593	0.0036	0.0056	0.5593
Adaboost	0.3171	0.0089	0.0001	0.3777
Random Forest	0.6459	0.0034	0.0005	0.6459
Xgboost	0.0792	0.0059	0.0001	0.0792

Table 4: Results after Wavelet analysis and standardized data

Method	Explained variance score	Mean absolute error	Mean squared error	R2 score
Decision Tree	0.7176	0.5587	0.9299	0.7203
Adaboost	0.005	0.4862	0.6282	0.0044
Random Forest	0.1655	0.4691	0.5252	0.1650

Table 5: Results after stacked auto encoder

6 Task 3: Recurrent neural network using long-short term memory

Author: Yingzhe Luo

6.1 Model principle

LSTM is characterized by the addition of layers of valve nodes in addition to the RNN structure.There are three types of valves: forget gate, input gate and output gate.These valves can be opened or closed to add to the current layer's calculations whether the output of the model network's memory state (the state of the previous network) at that layer has reached a threshold. The valve node uses sigmoid function to calculate the memory state of the network as input.If the output reaches the threshold value, the valve output is multiplied by the calculation result of the current layer as the input of the next layer. If the threshold is not reached, the output is forgotten. Weights for each layer including valve nodes are updated during each model back-propagation training

6.2 Results show

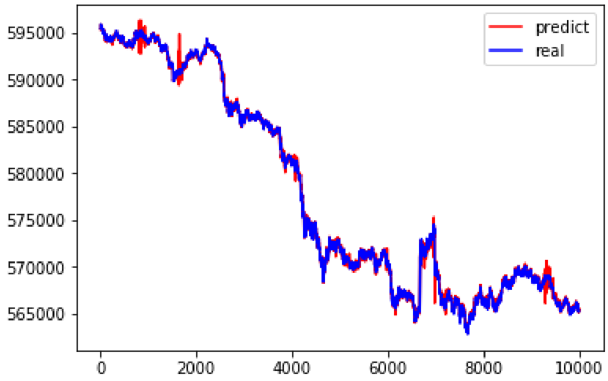

Figure 2: Predict VS Real

The red line is the prediction and the blue line is the real value. The first 8000 data is the prediction for train data, and the last 2000 data is the prediction of test data.

6.3 Batch size selection

If batchsize is too small, the algorithm will be difficult to converge. With the increase of batchsize, the speed of processing the same amount of data will increase rapidly, and the number of epoch needed to achieve the same accuracy will become more and more. Due to the contradiction between the two factors above, BatchSize increases to a certain point and reaches the optimal time. In my program, when the batch size is greater than 1000, the loss will not converge within 30 epoches. When the batch size is less than 50, the training time will be obviously lengthened and will converge within 5 epoches, which may cause some bad results. Finally, I choose 200 as the batch size of my program.

6.4 Hidden size

When the training set is determined, the number of input layer nodes and output layer nodes are then determined. First of all, a very important and difficult problem is how to optimize the number of hidden layer nodes and hidden layers.Experiments show that if the number of hidden nodes is too small, the network cannot have the necessary learning ability and information processing ability.On the contrary, if it is too much, it will not only greatly increase the complexity of the network structure (which is especially important for the network implemented by hardware), but also make the network fall into local minima more easily in the learning process and slow the learning speed of the network. The LSTM neural network we built is a layer with 32 neurons. To start with, initialize weights and allow all candidates to allow them to make an informed decision.

6.5 Dropout

Dropout is the temporary deletion of some neurons and gradient descent to update the weights of other neurons.Then the next time you temporarily delete other neurons do the same thing, you can prevent overfitting.Dropout provides a simple way to improve performance.Make sure you train your network a little longer.Dropout network will work better and better over time.

6.6 Learning rate selection

I starts with a normal-sized learning rate (LR) and shrinks towards the end point.Use a sub-set of training sets that do not train to determine when to reduce the learning rate and when to stop training.If you find a bottleneck in the validation set, divide LR by 2 and continue.Eventually, the LR will be very small, and it's time to stop training.This ensures that training data is not overfitted when verifying performance is compromised.

6.7 Optimizer selection

After consulting some materials on the Internet, I learned that Adam algorithm has the advantages of high computing efficiency, low memory requirements, and is suitable for solving optimization problems involving large-scale data and parameters, etc. Considering the running speed of personal computers and over 100,000 training samples in this project, I chose Adam algorithm.